(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 935 234 A1

# (12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
**11.08.1999 Bulletin 1999/32**

(51) Int. Cl.[6]: **G10H 1/00**

(21) Application number: **99102147.8**

(22) Date of filing: **03.02.1999**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**
Designated Extension States:
**AL LT LV MK RO SI**

(30) Priority: **05.02.1998 JP 3793198**

(71) Applicant:
**Casio Computer Co., Ltd.**
**Shibuya-ku, Tokyo 151-8543 (JP)**

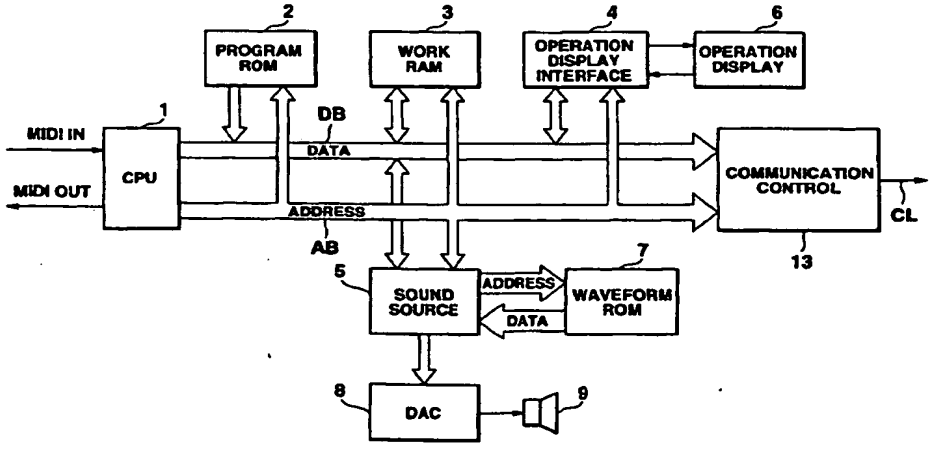
(72) Inventor:
**Kato, Hitoshi**
**c/o Casio Computer Co., Ltd.**
**Hamura-shi, Tokyo 205-8555 (JP)**

(74) Representative:
**Grünecker, Kinkeldey,**
**Stockmair & Schwanhäusser**
**Anwaltssozietät**
**Maximilianstrasse 58**
**80538 München (DE)**

## (54) Musical performance training data transmission

(57) A musical performance training system comprising a transmitter which transmits training note data for training musical performance, and a receiver which receives the training note data from the transmitter and instruct a trainee to train a musical performance based on the received training note data. The transmitter produces note data based on a musical performance, for example, at a keyboard. The note data is converted to training note data by reducing velocity data contained in the produced note data (S121, S234), shifting pitch data in units of an octave (S124, S239), and changing the timbre and pitch data to respective particular ones (S125, S250). If the note data is produced by a special key operation (J333, J334, J360-J365), control data which instructs the trainee to perform a pedal operation is added to the converted note data (J345-J349, J366-J369). The receiver displays or audibly outputs a performance instruction based on the received training note data. The training note data may be transmitted/received via a predetermined storage device on network (S206, S316, J206, J306).

**FIG.1**




EP 0 935 234 A1

[0014] According to this arrangement, the transmitter side is capable of freely transmitting to the receiver the musical note data produced by the performance as the training musical note data. The receiver can obtain training musical note data necessary for performance practicing even though the receiver has no training musical note data therein. In addition, the receiver is capable of receiving data for training the performance even thought the receiver has no hardware dedicated to training the performance.

FIG. 1 is a block diagram of a system for each embodiment of the present invention;
FIG. 2 is a plan view of an operation display panel of FIG 1;
FIG 3 is a main flow chart of operation of a transmitter of the first embodiment;
FIG 4 is a flow chart of an interrupt in the main flow chart of FIG. 3;
FIG 5 is a flow chart of a musical note producing/muting process of FIG. 3;
FIG 6 is a flow chart of a musical note producing process performed in a note-on time period of FIG. 5;
FIG 7 is a flow chart of a musical note muting process performed in a note-off time period of FIG. 5;
FIG. 8 is a main flow chart of reception of a reception end of the first embodiment;
FIG. 9 is a flow chart of a MIDI process of FIG. 8;
FIG. 10 is a flow chart of a key guide process of FIG. 8;
FIG. 11 is a flow chart of a keyboard process of FIG. 8;
FIG. 12 is a flow chart of a musical note producing process of FIG. 8;
FIG. 13 is a main flow chart of reception of a reception end of a modification of the first embodiment;
FIG. 14 is a flow chart of a musical note producing process of FIG. 13;
FIG. 15 is a main flow chart of transmission of a transmission end of a second embodiment;
FIG. 16 is a flow chart of a switch process of FIG. 15;
FIG. 17 is a flow chart of a mode switch process of FIG. 16;
FIG. 18 is a flow chart of a start/stop switch process of FIG. 16;
FIG. 19 is a flow chart of a time interrupt process of FIG. 16;
FIG. 20 is a flow chart of a transmission switch process of FIG. 15;
FIG. 21 is a flow chart of a training data creating process of FIG. 13;
FIG. 22 is a flow chart of a musical note producing process of FIG. 13;
FIG. 23 is a flow chart of a transmission process of FIG. 13;
FIG. 24 is a main flow chart of reception of a reception end of the second embodiment;
FIG. 25 is a flow chart of a switch process of FIG. 24;
FIG. 26 is a flow chart of a key guide switch process of FIG. 25;
FIG. 27 is a flow chart of a timer interrupt process of FIG. 25;
FIG. 28 is a flow chart of a reception process of FIG. 25;
FIG. 29 is a flow chart of a key guide process of FIG. 24;
FIG. 30 is a flow chart of a portion of the key guide process continuing from FIG. 29;
FIG. 31 is a flow chart of a reception process of FIG. 24;
FIG. 32 is a main flow chart of reception of a reception end of a modification of the second embodiment;
FIG. 33 is a flow chart of a switch process of FIG. 32;
FIG. 34 is a flow chart of a key guide switch process of FIG. 33;
FIG. 35 is a flow chart of the training data reading process of FIG. 32;
FIG. 36 is a flow chart of a portion of the training data reading process continuing from FIG. 35;
FIG. 37 is a flow chart of a mode switch process of a transmission end of a third embodiment;
FIG. 38 is a flow chart of a training data creating process of the transmission end of the third embodiment;
FIG. 39 is a flow chart of operation of a reception end of the third embodiment;
FIG. 40 is a flow chart of a switch process of FIG. 39;
FIG. 41 is a flow chart of a pedal data switch process of FIG. 40;
FIG. 42 is a flow chart of a key guide process of FIG. 39;
FIG. 43 is a flow chart of a pedal data switch process of FIG. 39;
FIG. 44 is a flow chart of a portion of the pedal data switch process continuing from FIG. 43;
FIG. 45 is a flow chart of a portion of the pedal data switch process continuing from FIG. 43;
FIG. 46 is a flow chart of a portion of the pedal data switch process continuing from FIG. 45;
FIG. 47 is a flow chart of operation of a transmission end of a modification of the third embodiment; and
FIG. 48 is a flow chart of a switch process of FIG. 47;

[0015] A first-a third embodiment of the present invention and a modification directed to an electronic keyboard instrument will be described next with reference to FIGS. 1-48.

[0016] In FIG. 1, a CPU 1 controls the electronic musi-

tion of the practicing data is on (step S119). If the instruction included in the MIDI data is to mute a musical note, the CPU 1 performs a note-off process or a musical note muting process and a conversion data producing process when the production of the practicing data is on (step S120). After the note-on or-off process, the CPU 1 terminates the FIG. 5 process.

**[0026]** FIG. 6 shows a flow of the note-on process in step S119 of the musical note producing/muting process of FIG. 5. In this process, the CPU produces MIDI send data which includes a changed velocity of the appropriate musical note data, and transfers it to a MIDI buffer (step S121). The CPU then determines whether the transmission system (sender) is in a silence mode in which no musical note is produced or a note-on mode (step S122). If the transmission system is in the silence mode, the CPU terminates the FIG. 6 process immediately. If the transmission system is in the note-on mode, the CPU determines whether the transmission system is in an ordinary note mode in which an ordinary musical note is produced or a timing note mode in which a timing note such as a click or castanets musical note is produced (step S123).

**[0027]** If the transmission system is in the ordinary note mode, the CPU then creates musical note producing MIDI data in which the pitch is shifted upward (toward a higher pitch) or downward (to a lower pitch) in units of an octave and transfers this data to the MIDI buffer (step S124). If the transmission system is in the timing note mode, the CPU produces musical note producing MIDI data in which a timbre of the produced musical note is assigned to the timing note, and transfers it to the MIDI buffer (step S125). After transfer of the musical note producing MIDI data in step S124 or S125, the CPU performs an original musical note producing process based on original or unchanged note control data (step S126), and then terminates the FIG. 6 process.

**[0028]** FIG. 7 shows the note-off process in step S120 of the FIG. 5 musical note producing/muting process. In this process, the CPU produces MIDI send data to mute a musical note involved in the appropriate musical note data and transfers it to the MIDI buffer (step S 127). The CPU then determines whether the transmission system is in the silence mode or in the note-on mode (step S128). If the transmission system is in the silence mode, the CPU terminates the FIG. 7 process immediately. If the transmission system is in the note-on mode, the CPU determines whether the transmission system is in the ordinary note mode in which the ordinary musical note is produced or in the timing note mode where a timing note such as a click or castanets note is produced (step S129).

**[0029]** When the transmission system is in the ordinary note mode, the CPU produces musical note muting MIDI data where the pitch is shifted upward (to a higher pitch) or downward (to a lower pitch) in units of an octave, and transfers this data to the MIDI buffer (step S130). If the transmission system is in the timing note mode, the CPU produces musical note muting MIDI data where a timbre of the produced musical note is assigned to a timing note, and transfers this data to the MIDI buffer (step S131). After transfer of the musical note muting MIDI data in step S130 or S131, the CPU performs the original musical note muting process based on the original or unchanged note control data (step S132), and then terminates the FIG. 7 process.

**[0030]** The receiver receives the MIDI data from the transmitter and produces note control data. FIG. 8 shows a main flow of operation of the performance training apparatus at the reception end. After the predetermined initialization (step J101), the CPU performs a switch process (step J102), a MIDI process (step J103), a key guiding process (step J104), a keyboard process (step J105), and a musical note producing process (step J106). The CPU then determines whether a manual power-off operation is performed (step J107). If not, the CPU performs processes in step J102-J106. If so in step J107, the CPU performs a power-off process (step J108) and then terminates the FIG. 8 process.

**[0031]** FIG. 9 shows a MIDI process in step J103 of the main flow of FIG. 8. In this process, the CPU determines whether MIDI data is received from an external device on a real time basis (step J109). If not, the CPU returns its control to the main flow. If so in step J109, the CPU determines whether the MIDI data is for guiding the performance training (step J110). When the MIDI data is received, the CPU stores the MIDI data in the registers GEVELT and GVELOCITY (step J111). Then, the CPU sets a guide flag GF at 1 (step J112), and then returns its control to the main flow.

**[0032]** When the CPU determines in step J111 that the received data is not the guiding MIDI data, the CPU determines whether the received data is musical note producing MIDI data (step J113). If so, the CPU stores the MIDI data in the registers SEVENT and SVELOCITY (step J114). The CPU then sets a musical note producing flag at 1 (step J115), and then returns to its control to the main flow. If the received data is neither the guiding MIDI data nor the musical note producing MIDI data, the CPU returns its control to the main flow.

**[0033]** FIG. 10 shows the key guide process in step J104 of the main flow of FIG. 8. In this process, the CPU determines whether a guide flag GF is at 1 (step J116). If not, the CPU returns its control to the main flow. If the GF is at 1, the CPU determines whether the MIDI data in GEVELT is note-on data (step J117). If so, the CPU turns on an LED of a key corresponding to musical note data in a register GEVENT (step J118). If the MIDI data is note-off data, the CPU turns off an LED for the key to the musical note data in the GEVENT (step J119). After turning on or off the LED, the CPU resets the GF at 0 (step J129), and then returns its control to the main flow.

**[0034]** FIG. 11 shows the keyboard process in step E105 of the main flow of FIG. 8. In this process, the CPU determines whether there is a key change (step

the CPU instructs the sound source to produce a musical note based on the musical note and velocity data (step J150). If the data in the EVENT is note-off data, the CPU instructs the sound source to mute the musical note based on the musical note data (step J151). Then, the CPU resets the HF at 0 (step J152).

**[0046]** Then, or if the HF is at 0 in step J148, the CPU 1 determines whether the GF is at 1 (step J153). If so, the CPU determines whether the data in the GEVENT is note-on data (step J154). If so, the CPU instructs the sound source to produce a musical note based on the musical note data in the GEVENT and velocity data in the GVELOCITY (step J155). If the data in the GEVENT is note-off data, the CPU instructs the sound source to mute the musical note based on the note-off data (step J156). After step J155 or J156, the CPU resets the GF at 0 (step J157).

**[0047]** The CPU 1 then determines whether the SF is at 1 (step J158). If so, the CPU 1 determines whether the data in the SEVENT is note-on data (step J159). If so, the CPU determines whether the data in the SEVENT is musical note data or tone data (step J160). If it is musical note data, the CPU instructs the sound source to produce a musical note based on the musical note data in the SEVENT and velocity data in the SVELOCITY (step J161). If it is tone data, the CPU instructs the sound source to produce a musical note based on the tone data in the SEVENT and velocity data in the SVELOCITY (step J162). After step J161 or J162, the CPU resets the SF at 0 (step J163) and then returns its control to the main flow.

**[0048]** If the data in the SEVENT is note-off data in step J159, the CPU determines whether the data in the SEVENT is musical note data or tone data (step J164). If it is musical note data, the CPU instructs the sound source to mute the musical note (step J165). If it is tone data, the CPU instructs the sound source to mute the tone (step J166). After step J165 or J166, the CPU resets the SF at 0 (step J163), and then returns its control to the main flow. If the GF is 0 in step J153 or if the SF is 0 in step J158, the CPU returns its control to the main flow.

**[0049]** Since the MIDI process in step J143 of the FIG. 13 main flow is the same as the MIDI process performed by the transmission end of FIG. 9, further description thereof will be omitted. Since the keyboard process in step J144 in the FIG. 13 main flow is the same as that performed by the transmission end of FIG. 11, further description thereof will be omitted.

**[0050]** A second embodiment will be described next. FIG. 15 shows a main flow operation of the transmission end in the second embodiment. After the predetermined initialization (step S201), the CPU performs a switch process (step S202), a keyboard process (step S203), a training data creating process (step S204), a musical note producing process (step S205), and a transmitting process (step S206). The CPU then determines whether a power-off process has been performed manually (step S207). If not, the CPU performs processing in steps S202-S206. If so in step S207, the CPU performs a power-off process (step S208), and then terminates the FIG. 15 flow.

**[0051]** FIG. 16 is a flow chart of the switch process in step S202 of the FIG. 15 main flow. In this process, the CPU performs a mode switch process (step S209), a start/stop process (step S210), a transmission switch process (step S211), and another switch process (step S212), and then returns its control to the main flow.

**[0052]** FIG. 17 is a flow chart of the mode process in step S209 of the FIG. 16 switch process. In this process, the CPU determines whether a creation mode switch has changed (step S213). If not, the CPU returns its control to the FIG. 16 flow. When the production mode switch changes from off to on, the CPU sets a mode flag MF at 1 (step S214). When the creation mode switch changes from on to off, the CPU resets the MF at 0 (step S215).

**[0053]** After step S214 or S215, the CPU determines whether the timing sound mode switch has changed (step S216). If not, the CPU returns its control to the FIG. 16 flow. If so in step S216, the CPU sets the timing flag TF at 1 (step S217). When the switch has changed from on to off, the CPU resets the TF at 0 (step S218). After step S217 or S218, the CPU returns its control to the FIG. 16 flow.

**[0054]** FIG. 18 is a flow chart of the start/stop process in step S219 in the FIG. 16 switch process. In this process, the CPU determines whether the start/stop switch is on (step S219). If not, the CPU returns its control to the FIG. 16 flow. If the switch is on, the CPU inverts a start flag STF (step S220). The CPU then determines whether the STF is at 1 (step S221). If so, the CPU then sets an address register AD at 0 (step S223), and then releases timer interrupt (step S224). If the STF at 0 in step S221, the CPU inhibits timer interrupt (step S225), and stores END data at a location MEM [AD] of the work RAM 3 which is specified by an address in the address register AD (step S226).

**[0055]** After the timer interrupt is released from its inhibition in step S224, or after the END data is stored in step S226, the CPU returns its control to the FIG. 16 flow. When the timer interrupt is released, the CPU increments the value of the register TIME each time a predetermined time elapses (step S227), and then returns its control to the FIG. 16 flow.

**[0056]** FIG. 20 is a flow chart of the transmission switch process in step S211 of the FIG. 16 switch process. In this process, the CPU determines whether the transmission switch has been turned on (step S228). If so, the CPU sets a transmission flag SOF at 1 (step S229). Thereafter or if the transmission switch is not turned on in step S228, the CPU returns its control to the FIG. 16 flow.

**[0057]** FIG. 21 is a flow chart of the training data producing process in step S204 of the FIG. 15 main flow chart. In this process, the CPU determines whether a

to the musical note data in the GEVNET (step J229). After turning on/off the LED, the CPU increments the AD (step J230).

**[0069]** The CPU then determines whether the timing flag TF is at 0 in FIG. 30 (step J232). If the TF is at 1, the CPU then determines whether the data in the GEVENT is note-off data (step J233). If the TF is 0 or if the data in the GEVENT is note-on data, the CPU then stores in the register SEVENT the data at the location MEM [AD] of the RAM specified by an address in AD (step J234). The CPU then increments the AD (step J235) and stores the data at the appropriate location MEM [AD] in the register SVELOCITY (step J236), and then sets the musical note production flag SF at 1 (step J237).

**[0070]** After setting the SF or if the data in the GEVENT is note-off data in step J233, the CPU increments the AD (step J238), determines whether the data at the location MEM [AD] is END data (step J239). If not, the CPU returns its control to the FIG. 24 main flow. If so in step J239, the CPU then resets the STF at 0 (step J240), inhibits the timer interrupt (step J241) and then returns its control to the FIG. 24 main flow.

**[0071]** The keyboard process in step J204 of the main flow is the same as the FIG. 11 flow in the first embodiment, and further description thereof will be omitted. The musical note producing process in step J205 of the main flow is the same as the FIG. 12 flow of the first embodiment, and further description thereof will also be omitted.

**[0072]** FIG. 31 is a flow chart of the reception process in step J206 of the main flow. In this process, the CPU 1 determines whether the reception flag ZF is at 1 (step J242). If the reception flag ZF is at 0, the CPU then returns its control to the FIG. 24 main flow. If the ZF is at 1, the CPU then transfers to the RAM the data received from the server on the network (step J243). The CPU then determines whether the data reception from the server has ended (step J244). If not, the CPU then continues the data transfer in step J243. If the reception has ended, the CPU then resets the ZF at 0, and then returns its control to the FIG. 24 main flow.

**[0073]** A modification of the process performed by the reception end in the second embodiment will be described next with reference to a main flow chart of FIG. 32. After the predetermined initialization (step J246), the CPU 1 performs a switch process (step J247), a keyboard process (step J248), a training data reading process (step J249), a musical note producing process (step J250), and a reception process (step J251). The CPU then determines whether a power-off operation has been performed manually (step S252) If not, the CPU performs processing in steps S247-S251. If so in step S252, the CPU performs a power-off process (step S253) and then terminates the FIG. 32 process.

**[0074]** FIG. 33 is a flow chart of the switch process at the FIG. 32 main flow. In this process, the CPU performs a key guide switch process (step 254), a reception switch process (step J255), and another switch process (step J256), and then returns its control to the FIG. 32 main flow.

**[0075]** FIG. 34 shows the key guide switch process in step J254 of the FIG. 33 switch process. In this process, the CPU 1 determines whether the key guide switch has been turned on (step J257). If not, the CPU returns its control to the switch process of FIG. 33. If the key guide switch has been turned on, the CPU then inverts the start flag STF (step J258) and then determines whether the STF is at 1 (step J259). If so, the CPU resets the time register TIME at 0 (step J260), sets the address register AD at 0 (step J261), and then releases the timer interrupt (step J262). If the STF is 0 in step J259, the CPU inhibits timer interrupt (step J263).

**[0076]** After releasing the timer interrupt in step J262, or the CPU inhibits the timer interrupt in step J263, the CPU returns its control to the FIG. 33 switch process. When the CPU releases the timer interrupt, the CPU increments the value in the TIME each time a predetermined time elapses, as shown in FIG. 27 in the second embodiment, and returns its control to the FIG. 33 switch process.

**[0077]** The reception switch process in step J255 in the FIG. 33 switch process is the same as the FIG. 28 flow of the second embodiment and further description thereof will be omitted. In the FIG. 32 main flow, the keyboard process in step J248 is the same as the flow of FIG. 11 of the first embodiment, and further description thereof will be omitted.

**[0078]** FIGS. 35 and 36 are a flow chart of the training data reading process in step J249 in the FIG. 32 main flow. In FIG. 35, the CPU determines whether the STF is at 1 (step J264). If so, the CPU determines whether the data in the TIME has reached the time data at the location MEM [AD] (step J265). If the STF is at 0, or if the data in the TIME has not reached the time data at the location MEM [AD], the CPU returns its control to the FIG. 32 main flow.

**[0079]** When the STF is at 1 in step J264 and the CPU determines in step J265 that the data in the TIME has reached the time data at the location MEM [AD], the CPU 1 increments the AD (step J266). The CPU then stores in the register GEVENT the data at the location MEM [AD] of the RAM specified by an address in the AD (step J267). The CPU then increments the AD (step J268) and sets the guide flag GF at 1 (step J269).

**[0080]** Then, the CPU 1 determines whether the TF is at 0 in FIG. 36 (step J270). If the TF is at 1, the CPU determines whether the data in the GEVENT is note-off data (step J271). If the TF is at 0, or if the data in the GEVENT is note-on data, the CPU then stores in the register SEVENT the data at the location MEM [AD] of the RAM specified by an address in the AD (step J272). Then, the CPU 1 increments the AD (step J273), stores data at the location MEM [AD] in the register SVELOCITY (step J274), and then sets the musical note producing flag SF at 1 (step J275).

switch process.

**[0094]** FIG. 42 is a flow chart of the key guide process in step J302 of the FIG. 39 main flow. In this process, the CPU determines whether the start flag STF is at 1 (step J315). If so, the CPU determines whether the data in the time register TIME has reached the time data at the location MEM [AD] of the RAM specified by an address in the AD (step J316). If the STF is at 0, or the data in the TIME has not reached the time data at the location MEM [AD], the CPU then returns its control to the FIG. 39 main flow.

**[0095]** When the STF is at 1 and the data in the TIME has reached the time data at the location MEM [AD], the CPU increments the AD (step J317). The CPU then determines whether the data at the location MEM [AD] is note-on data (step J318). If so, the CPU then turns on an LED for a key corresponding to the musical note data at the MEM [AD] (step J319). If the data at the MEM [AD] is not note-on data, the CPU determines whether the data at the MEM [AD] is note-off data (step J320). If so, the CPU turns off an LED for a key corresponding to the musical note data at the MEM [AD] (step J321).

**[0096]** If the data at the MEM [AD] data is neither note-on data nor note-off data, the CPU determines whether the data at the MEM [AD] is pedal-on data (step J322). If so, the CPU turns on a corresponding pedal mark (step J323). If the data at the MEM [AD] is not any of the note-on, note-off and pedal-on data, the CPU then determines whether the data at the MEM [AD] is the pedal-off data (step J324). If so, the CPU turns off a corresponding pedal mark (step J325).

**[0097]** After turning on/off the ELD or pedal mark, the CPU increments the AD (step J326). The CPU then determines whether the data at the MEM [AD] is time data (step J327). If not, the CPU shifts its control to step J318 to determine whether the data at the MEM [AD] is time data. If so, the CPU returns its control to the FIG. 39 main flow.

**[0098]** If the data at the MEM [AD] is neither musical note data nor pedal data in the FIG. 42 flow, the CPU determines whether the data at the MEM [AD] is END data (step J328). If it is END data, the CPU resets the start flag STF at 0 (step J329), inhibits the time interrupt (step J330), turns off the lighting LED and pedal mark (step J331), and then returns its control to the FIG. 39 main flow. If the data at the MEM [AD] is not END data in step J328, the CPU shifts its control to step J326 to increment the AD, and determines data at the MEM [AD] in step J327.

**[0099]** The musical note producing process in step J304 of the FIG. 39 main flow is the same as the FIG. 22 flow in the second embodiment and further description thereof will be omitted.

**[0100]** FIGS. 43-46 are a flow chart of the pedal data producing process in step J305 of the FIG. 39 main flow. In FIG. 43, the CPU determines whether the pedal flag PF is at 1 (step J332). If the PF is at 0, the CPU returns its control to the FIG. 39 main flow. If the PF is at 1, the CPU sets the AD at 0 (step J333). Then, the CPU determines whether the data at the location MEM [AD] is event data (step J334).

**[0101]** If the data at the MEM [AD] is not event data, the CPU increments the AD (step J335). The CPU then determines whether the data at the MEM [AD] is END data (step J336). If not, the CPU shifts its control to step J334, and determines whether the data at the MEM [AD] of the RAM specified by the incremented address in the AD is event data.

**[0102]** If so, the CPU determines whether the event data is note-on data (step J337). If so, the CPU determines whether the sostenute pedal flag SPF is at 0 (step J338). If so, the CPU sets at 0 a pointer N which indicates the number of a key for the preceding note-on data (step J339) and increments the N while determining whether the value of an on flag ONF (N) for the preceding musical note event data is at 1, which implies musical note production (step J340). If so, the CPU then determines whether the absolute value of the musical note data at the MEM [AD] minus the musical note data in the pointer N is larger than a predetermined value, that is, whether a key interval (or the number of keys) between a key corresponding to the last note-on data and a key corresponding to the preceding note-on data is larger than a predetermined value (step J341).

**[0103]** If the ONF (N) is at 0 (musical note muting) in step J340, or if the absolute value of the difference in musical note data key is smaller than the predetermined value in step J341, the CPU increments the N (step J342). The CPU then determines whether the N has exceeded the predetermined number (step J343). If not, the CPU returns its control to step J340 to determine the value of the ONF (N).

**[0104]** If the absolute value of the difference in musical note data key is larger than the predetermined value in step J341, the CPU sets the SPF at 1 (step J344). That is, if the interval (the number of keys) between the key corresponding to the last note-on data and the key corresponding to the preceding note-on data is larger than the predetermined value, the CPU sets a flag to turn on a mark indicative of the sostenute pedal.

**[0105]** The CPU then stores an address in the AD (N) in an address register PAD (step J345). In the FIG. 44 flow chart, the CPU decrements the respective data at PAD + 1 and subsequent addresses by two addresses, that is, empties two locations after the PAD of the preceding note-on event data (step J347), and then stores at a location MEM [AD + 1] (first empty area) the same time data as the preceding note-on time data TIME (N) (step J348). Then, the CPU stores pedal-on event data at a location MEM [AD + 2] (second empty area) (step J349).

**[0106]** Then, or when the SPF is at 1 in step J338 of FIG. 43, or if N has exceeded the predetermined number in step J343 of FIG. 43, the CPU sets N at 0, increments N while determining the value of the on-flag in step J350 of FIG. 44, that is, whether the ONF (N) is

step S310 of the FIG. 47 main flow. In this process, the CPU performs a mode switch process (step S318), a start/stop switch process (step S319), a transmission switch process (step S320), a pedal data producing switch process (step S321), and another switch process (step S322), and then returns its control to the main flow.

**[0119]** The keyboard process in step J312 of the FIG. 47 main flow is the same as the flow of FIG. 11 in the first embodiment. The training data producing process in step J313 is the same as the flow of FIG. 22 in the second embodiment. The pedal data producing process in step J314 is the same as the flow of FIGS. 43-46 in the third embodiment. The sound producing process in step J315 is the same as the flow of FIG. 23 in the second embodiment. The transmitting process in step J316 is the same as the flow of FIG. 21 in the second embodiment. Thus, further description of those processes will be omitted.

**[0120]** The mode switch process in step J319 of the FIG. 48 switch process is the same as that of FIG. 17 in the second embodiment. The start/stop switch process in step J320 is the same as the flow of FIG. 19 in the second embodiment. The transmission switch process in step J321 is the same as the flow FIG. 20 in the second embodiment. The pedal data creating switch process in step J322 is the same as the flow of FIG. 41 in the third embodiment. Thus, further descriptions of those processes will be omitted.

**[0121]** The flow charts of processes performed by the reception end in the modification of the third embodiment are the same as the flow charts of FIGS. 24-31 in the second embodiment, and further description thereof will be omitted.

## Claims

1. A performance training data transmitter comprising training note data producing means for producing training note data (S104, S121, S124, S125, S203, S234, S239, S240, S312-S314), and transmitting means for transmitting the training note data produced by said training note data producing means (S110, S206, S316), characterized in that:

   the training note data producing means comprises note data producing means for producing note data (S104, S203, S312); and data converting means for converting the note data produced by said note data producing means to training note data (S119, S120, S204, S313, S314).

2. The performance training data transmitter according to the claim 1, wherein said data converting means (S121, S234) converts the note data produced by said note producing means to training note data by reducing velocity data contained in the note data.

3. The performance training data transmitter according to the claim 1, wherein said data converting means (S124, S239) converts the note data produced by said note producing means to training note data by shifting pitch data contained in the note data in units of an octave.

4. The performance training data transmitter according to the claim 1, wherein said data converting means (S125, S240) converts the note data produced by said note producing means to training note data having a predetermines pitch unrelated to the pitch data contained in the note data and a timbre different from those of the note data.

5. The performance training data transmitter according to the claim 1, wherein said data converting means (S314) comprises data determining means (J333, J334, J360-J365) for determining whether keyboard performance based on the produced note data involves a special key depression, and control data producing means, responsive to said data determining means determining that the keyboard performance involves the special key depression, for producing control data which instructs a trainee to perform a pedal operation for supplementing the special key depression and for adding the control data to the note data (J345-J349, J366-J369).

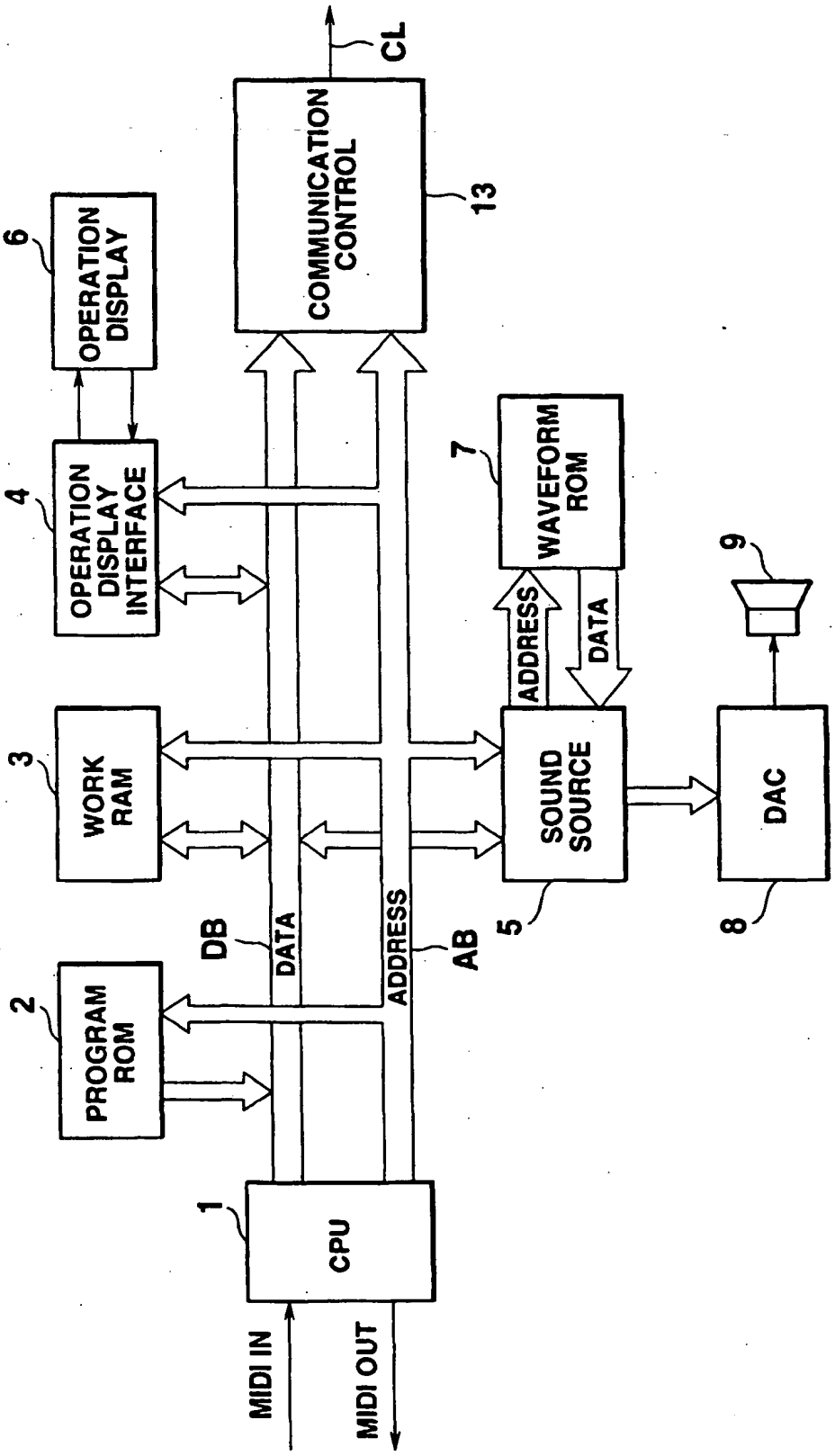
6. The performance training data transmitter according to the claim 1, wherein the special key depression is such that there are more than a predetermined number of keys between any particular depressed key and the next depressed key, and wherein the pedal operation is performed to continue note production by the particular depressed key.

7. The performance training data transmitter according to the claim 1, wherein said transmitting means (S206, S316) transmits the training note data to a predetermined storage device on a network at any time.

8. A performance training data receiver comprising receiving means for receiving the training note data from an external device (J109, J110, J206, J223-J227, J232-J237, J249), training instructing means for instructing a trainee to train musical performance on the basis of the training note data received by said receiving means (J104, J106, J145, J203), characterized in that:

   said receiving means comprises note data receiving means for receiving note data sent by the external device (J109, J206, J306); and

FIG.1



# FIG.3

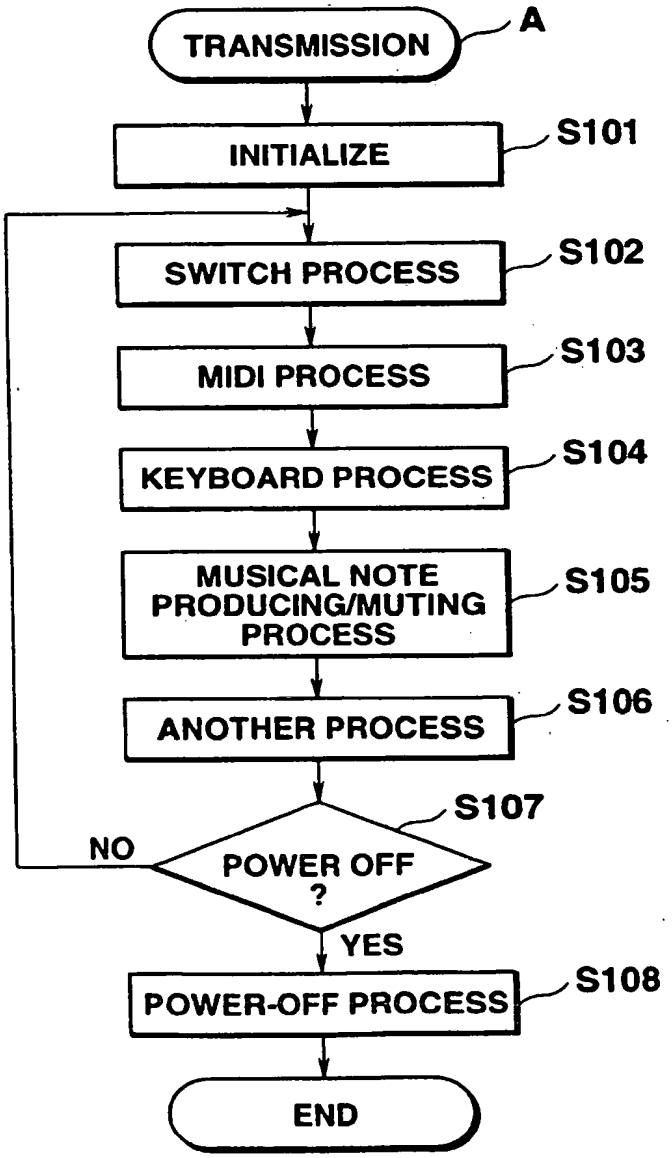


FIG.5

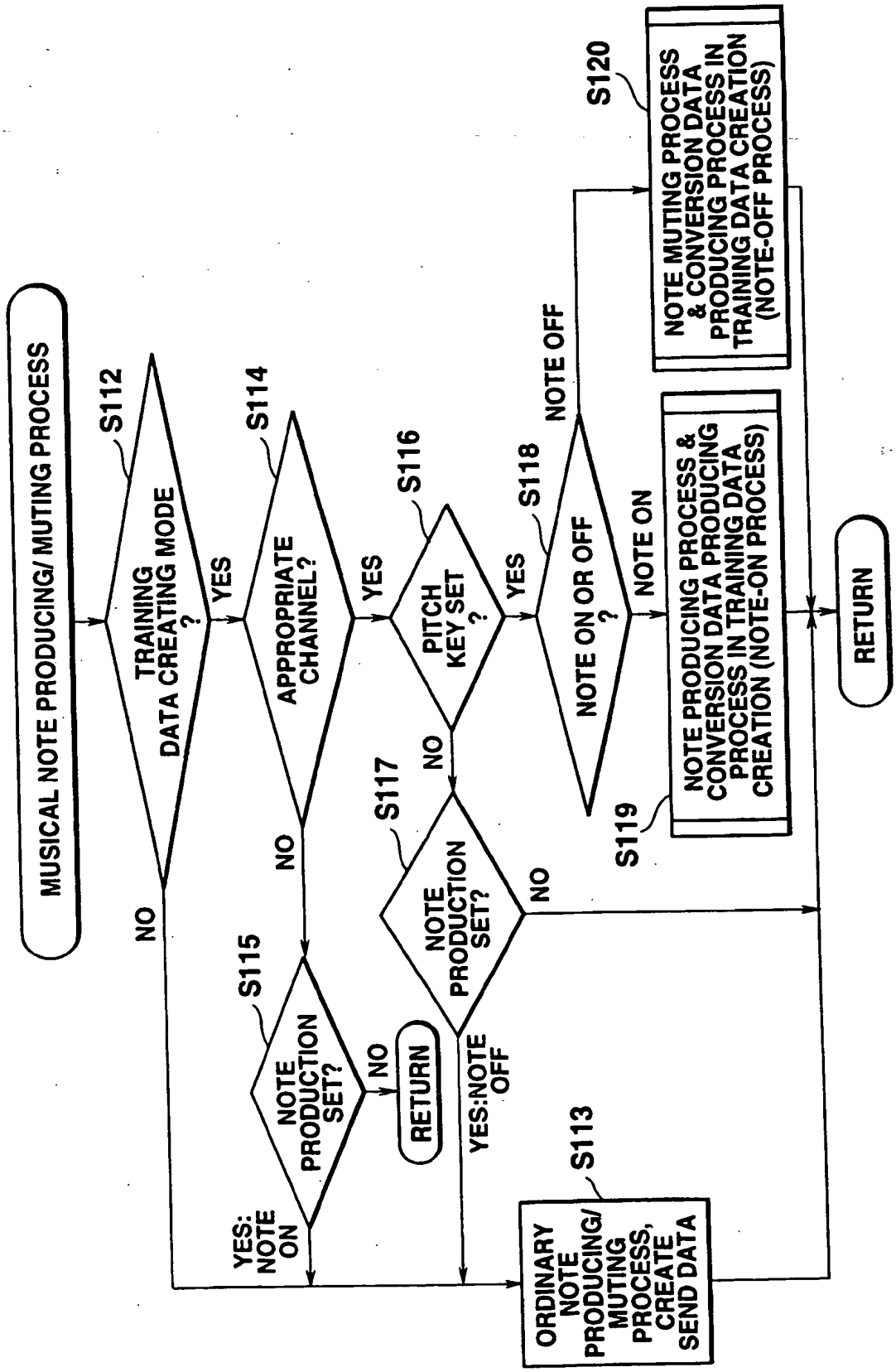


FIG.7

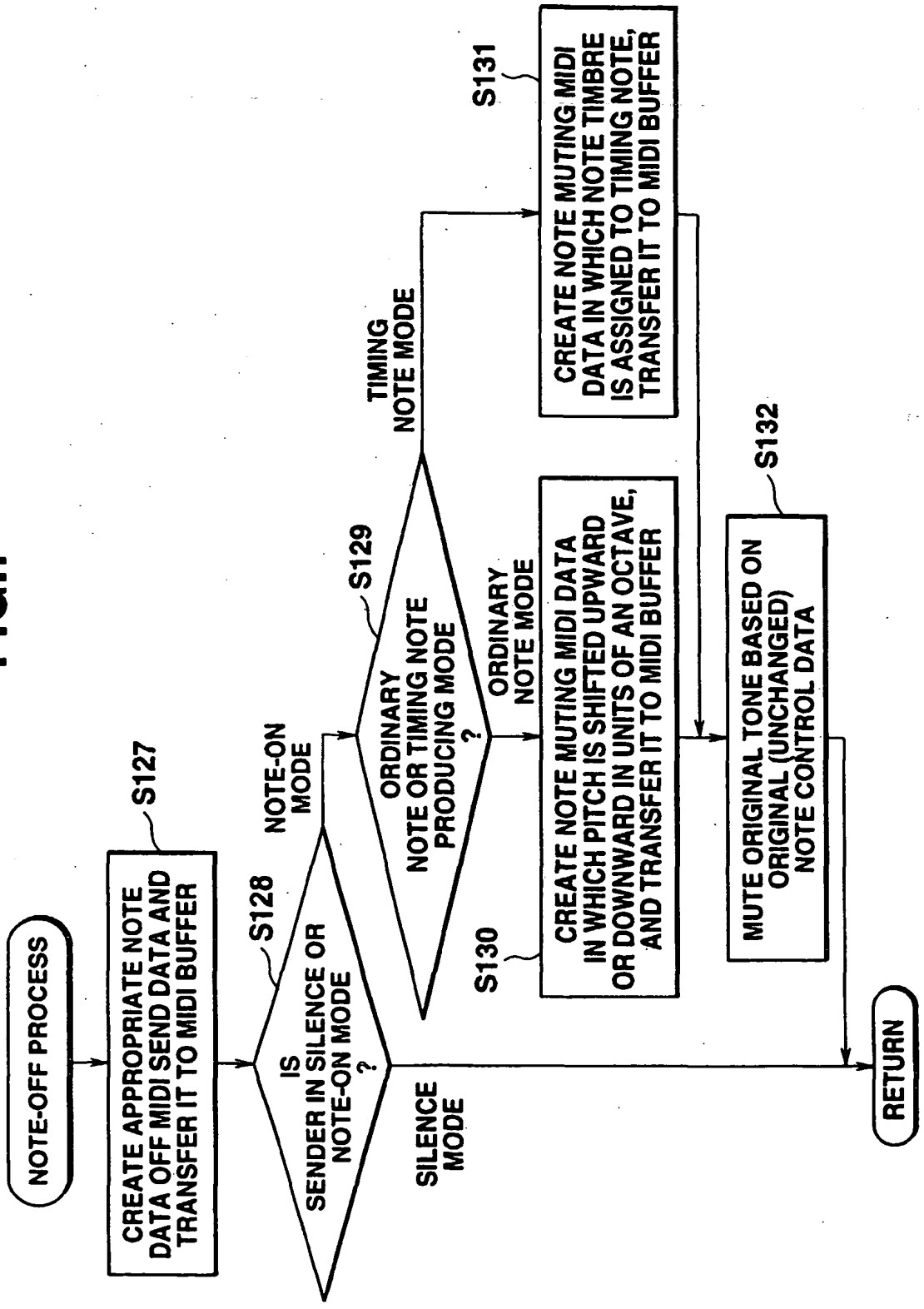
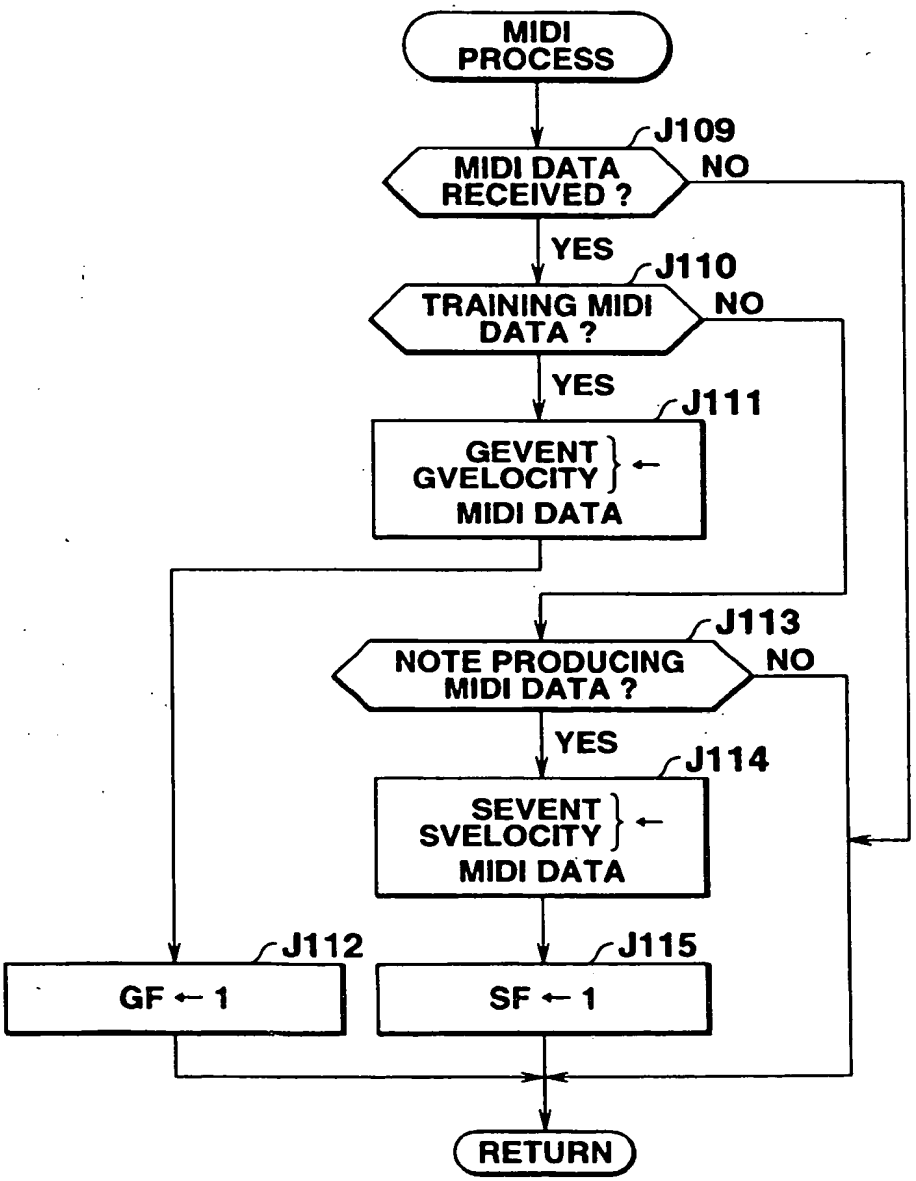
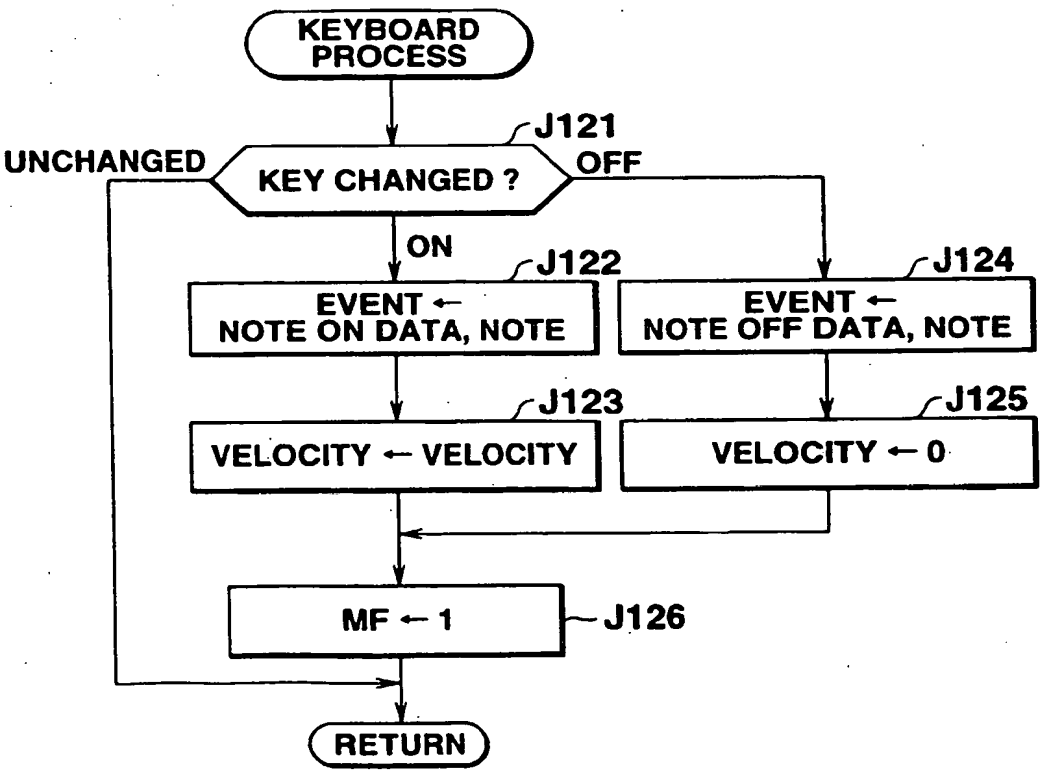


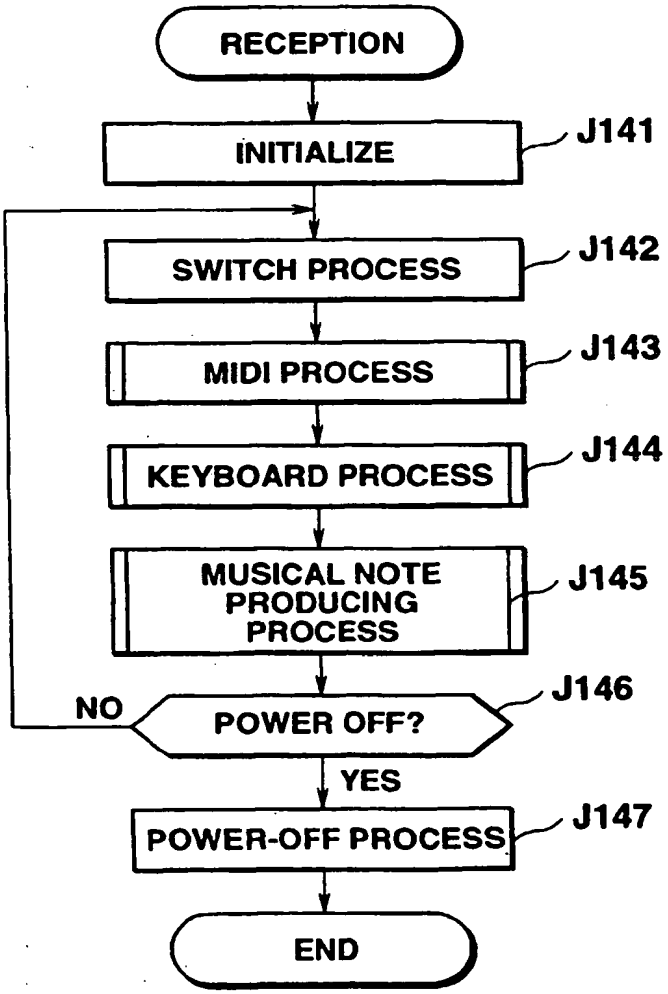
FIG.9



# FIG.11



# FIG.13



# FIG.15

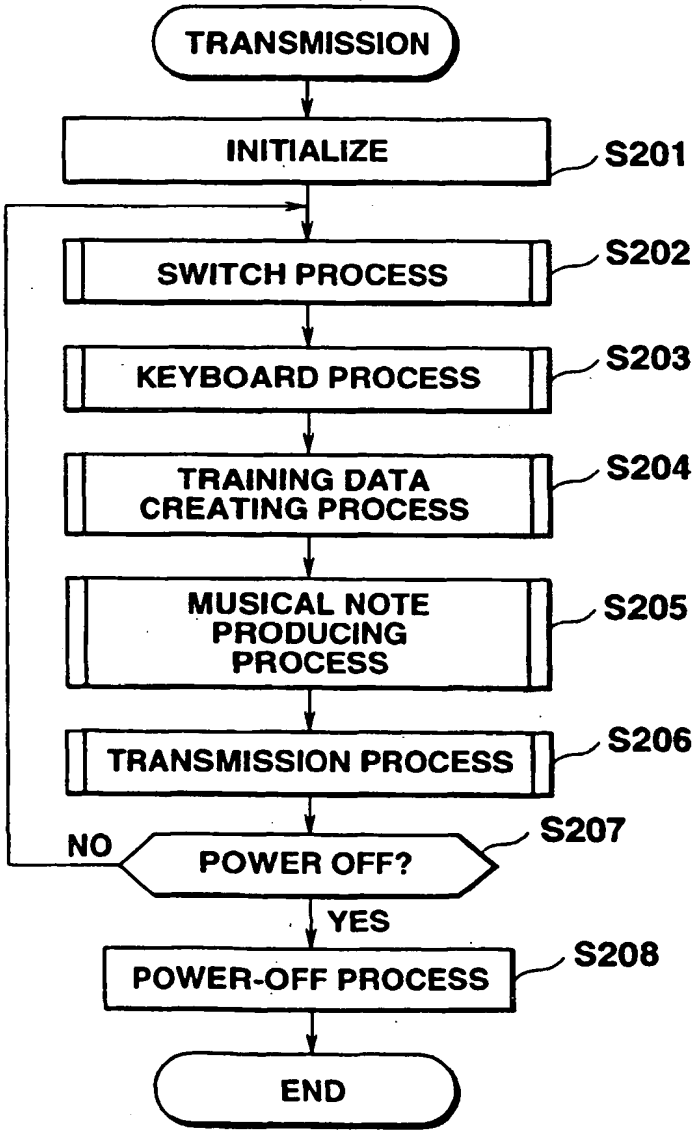


FIG.17

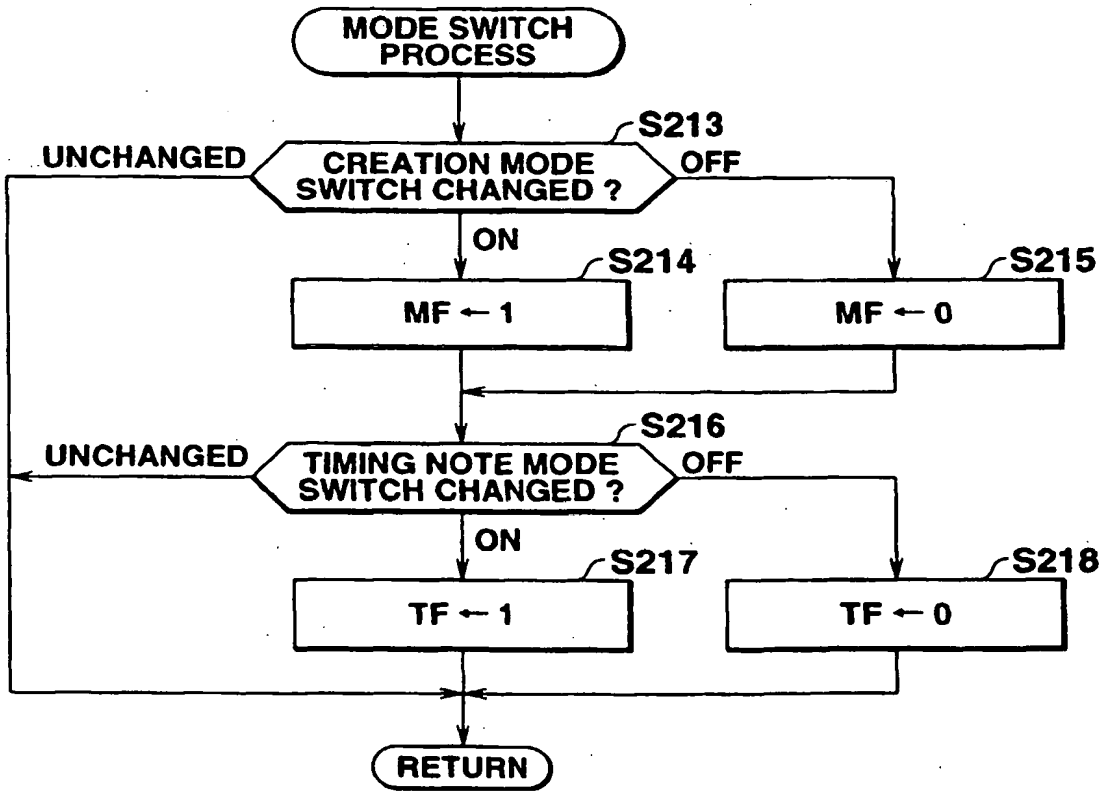


FIG.19

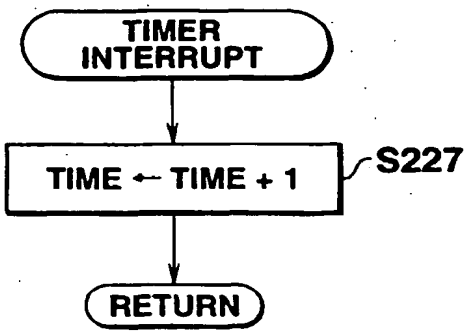



FIG.20

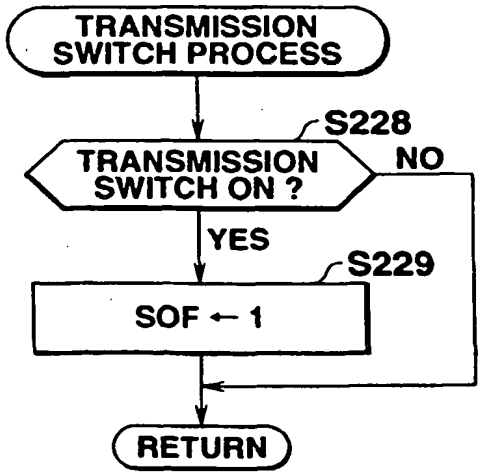


FIG.22

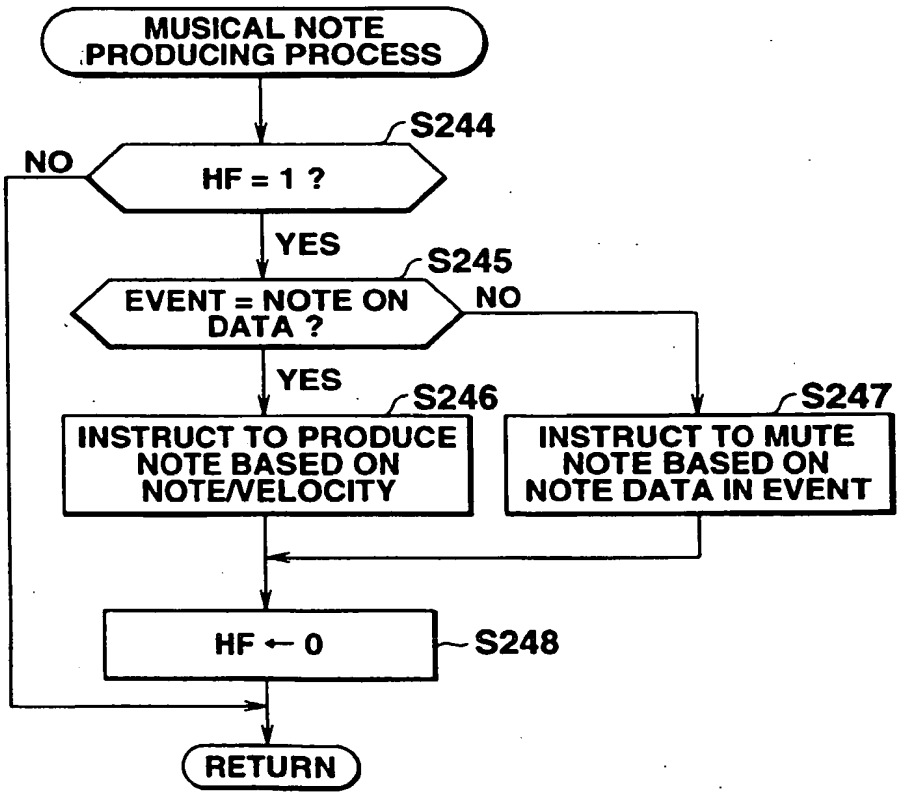
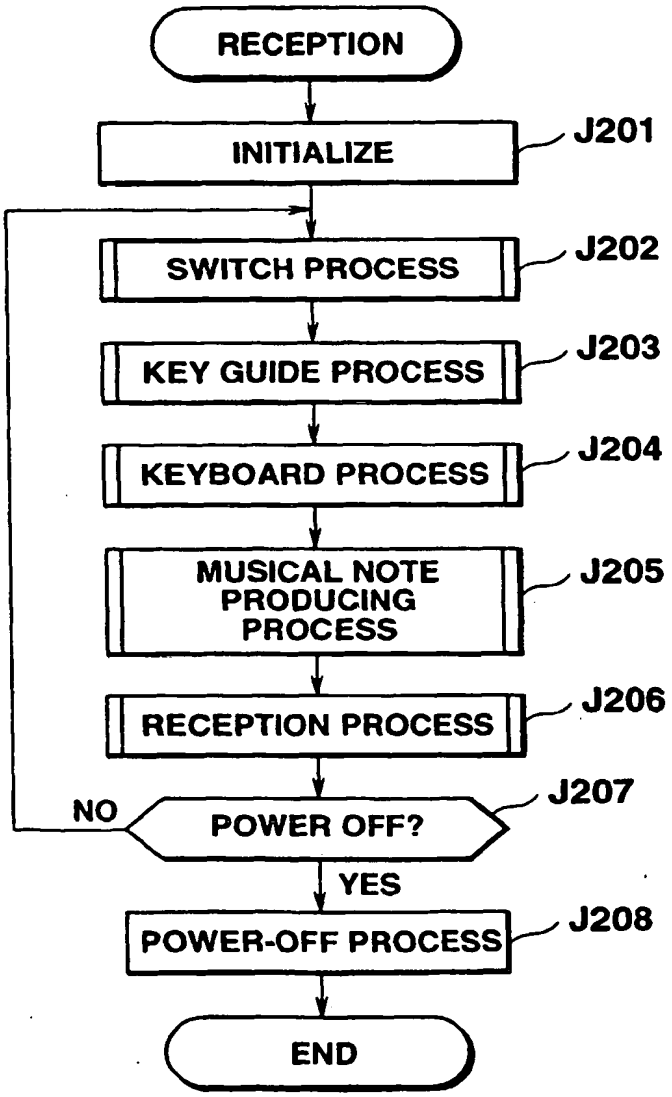
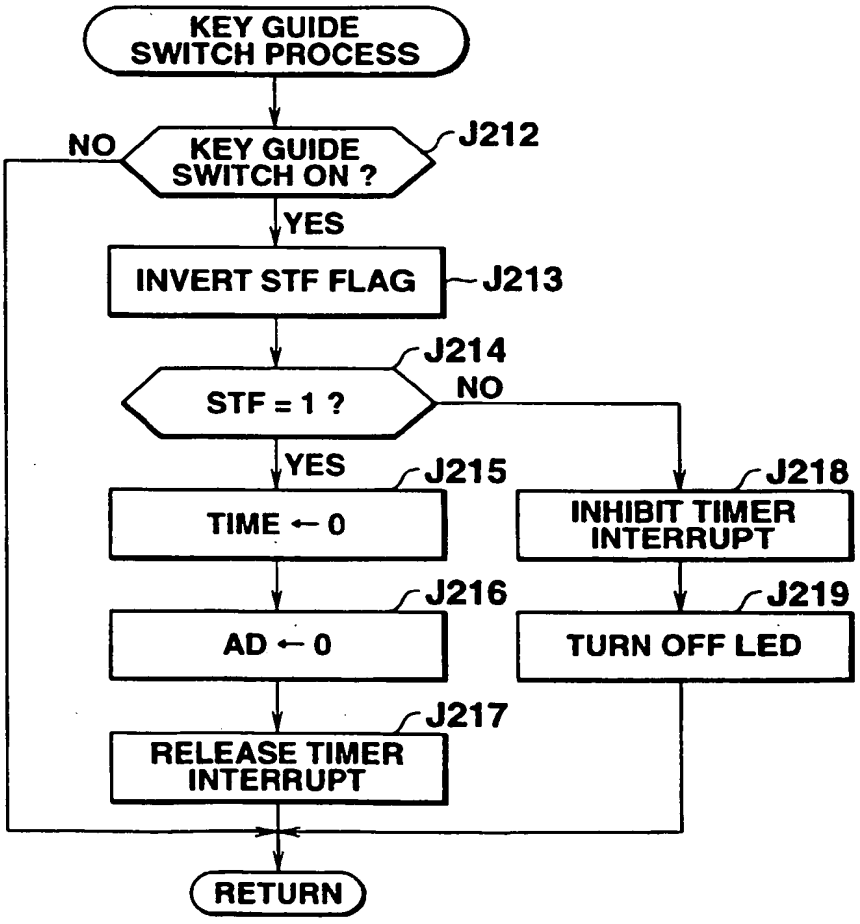


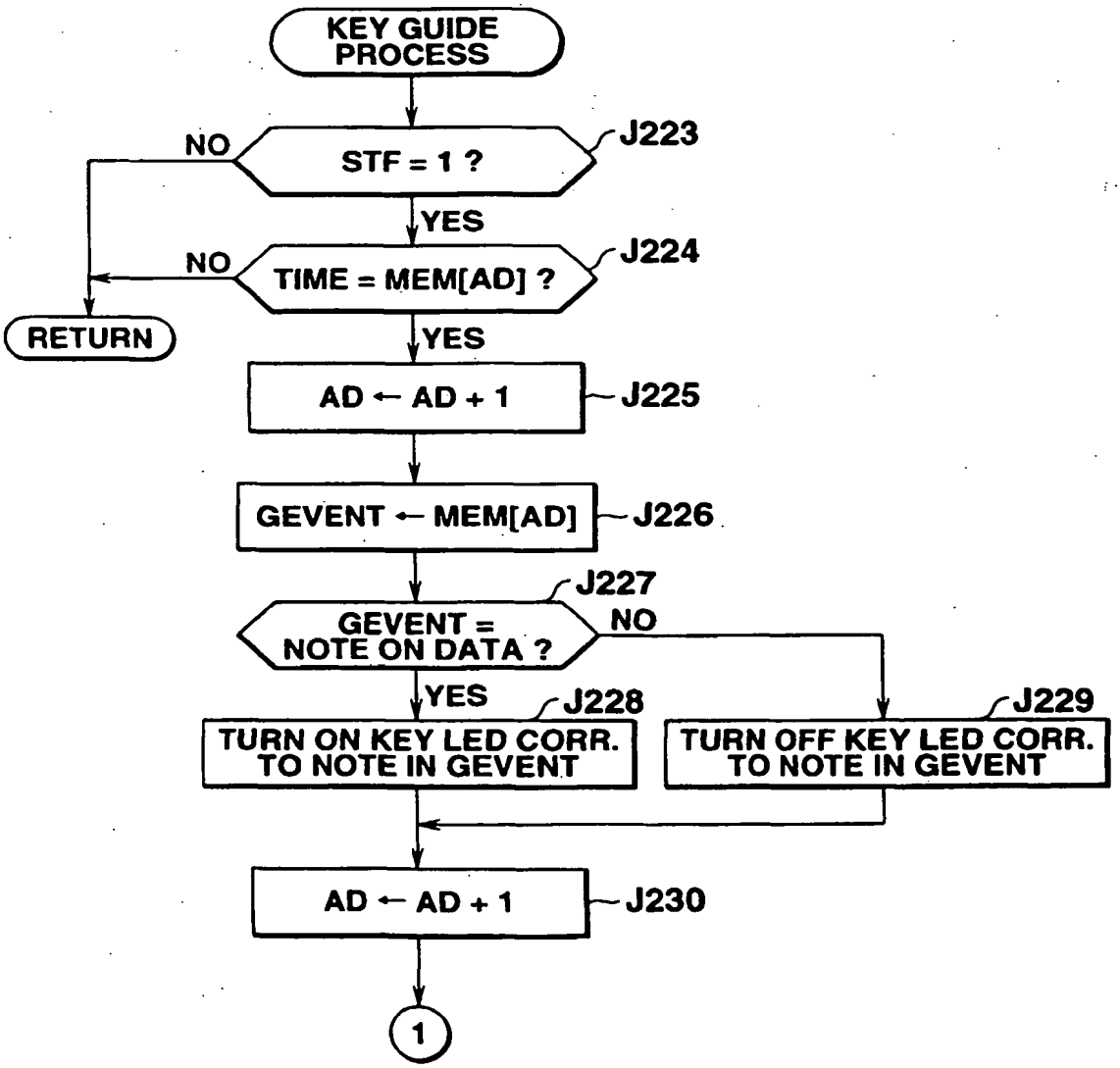
FIG.24



# FIG.26



# FIG.29



# FIG.31

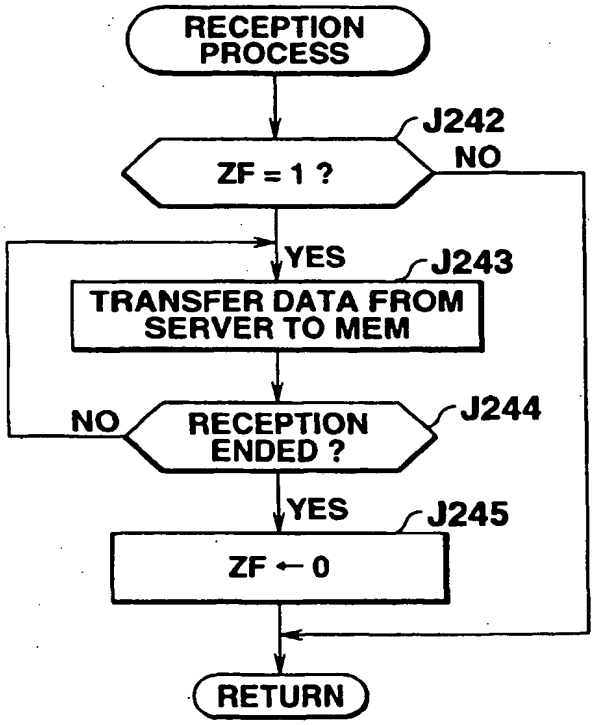
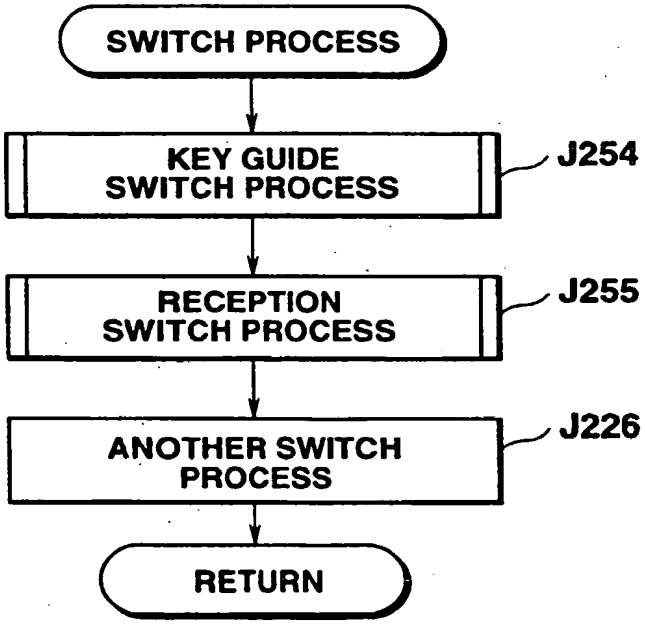


FIG.33



# FIG.35

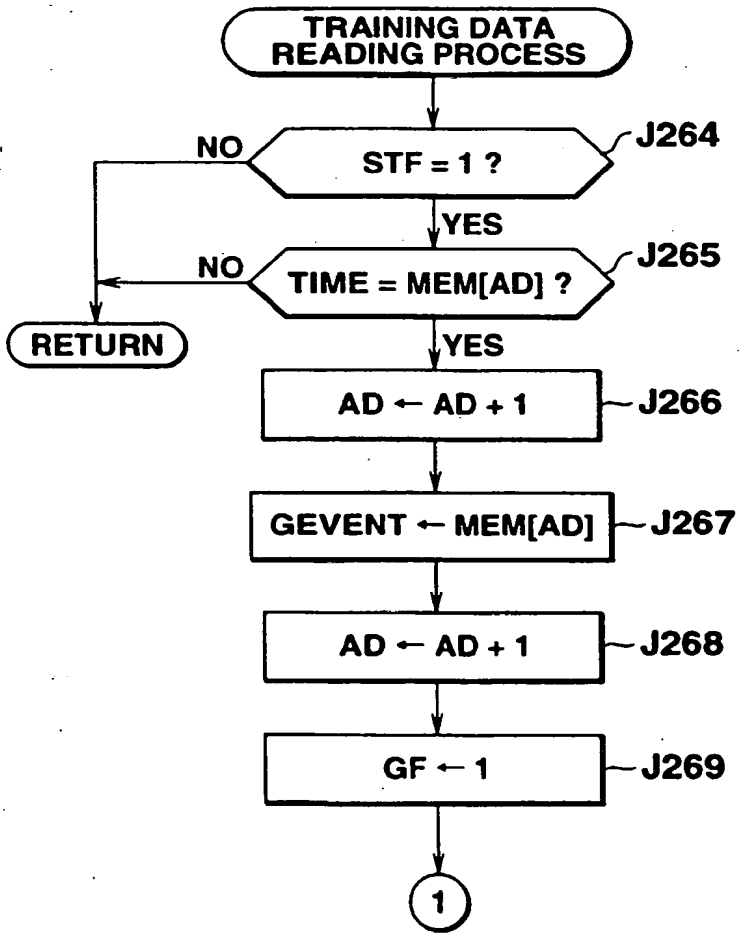


FIG.37

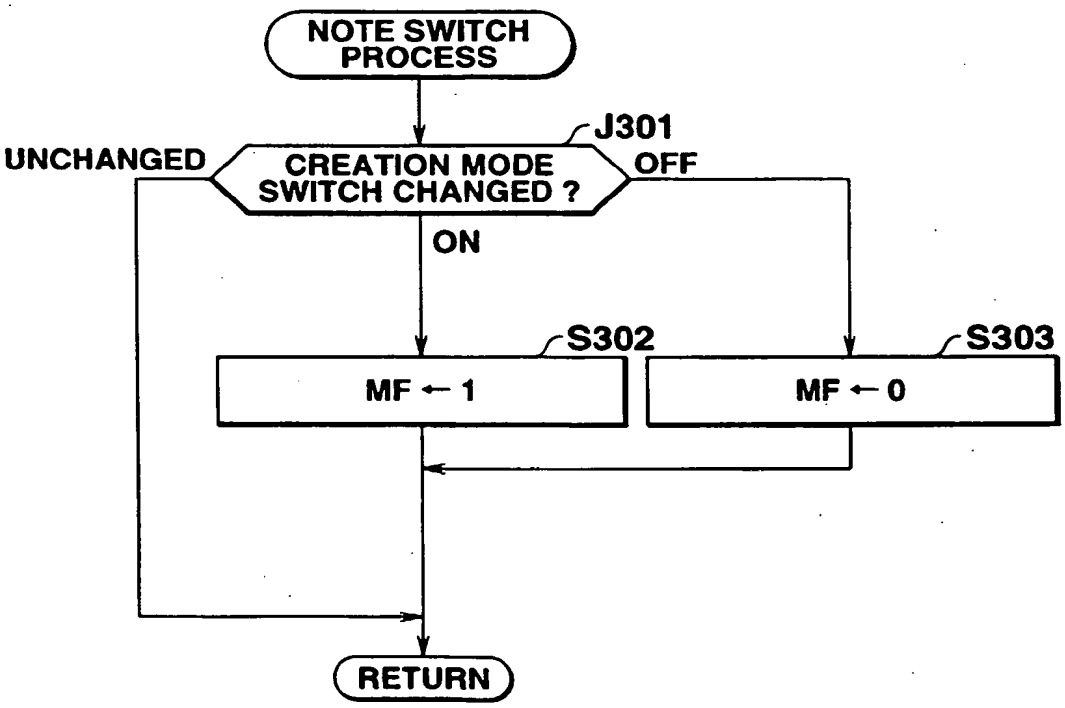
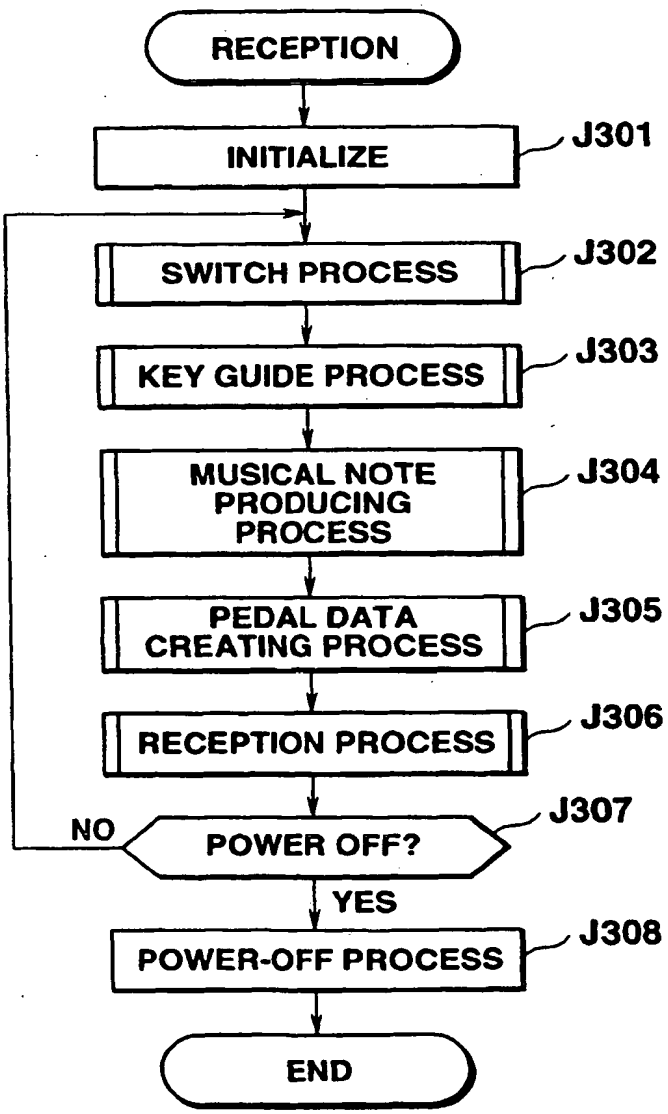
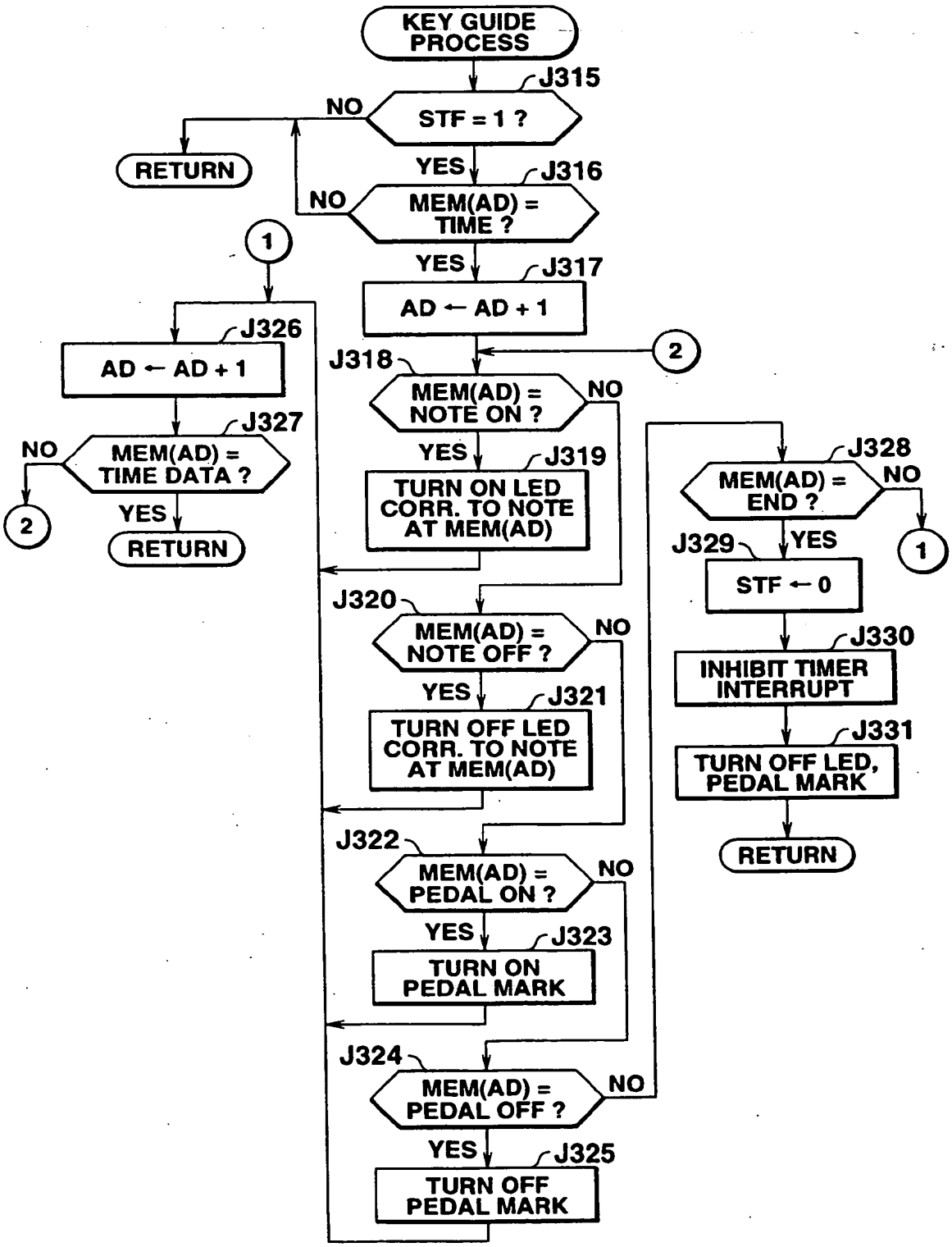


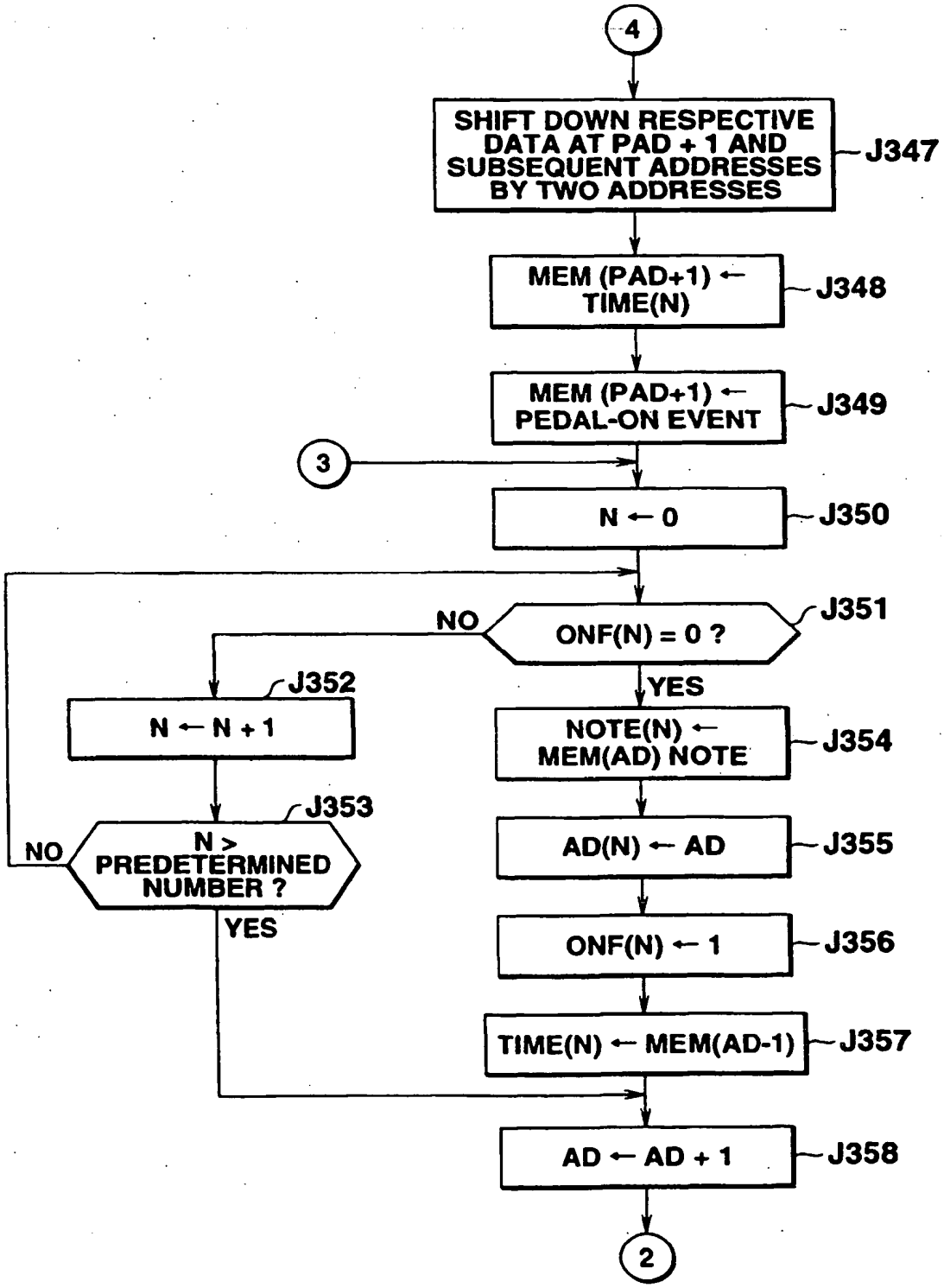
FIG.39



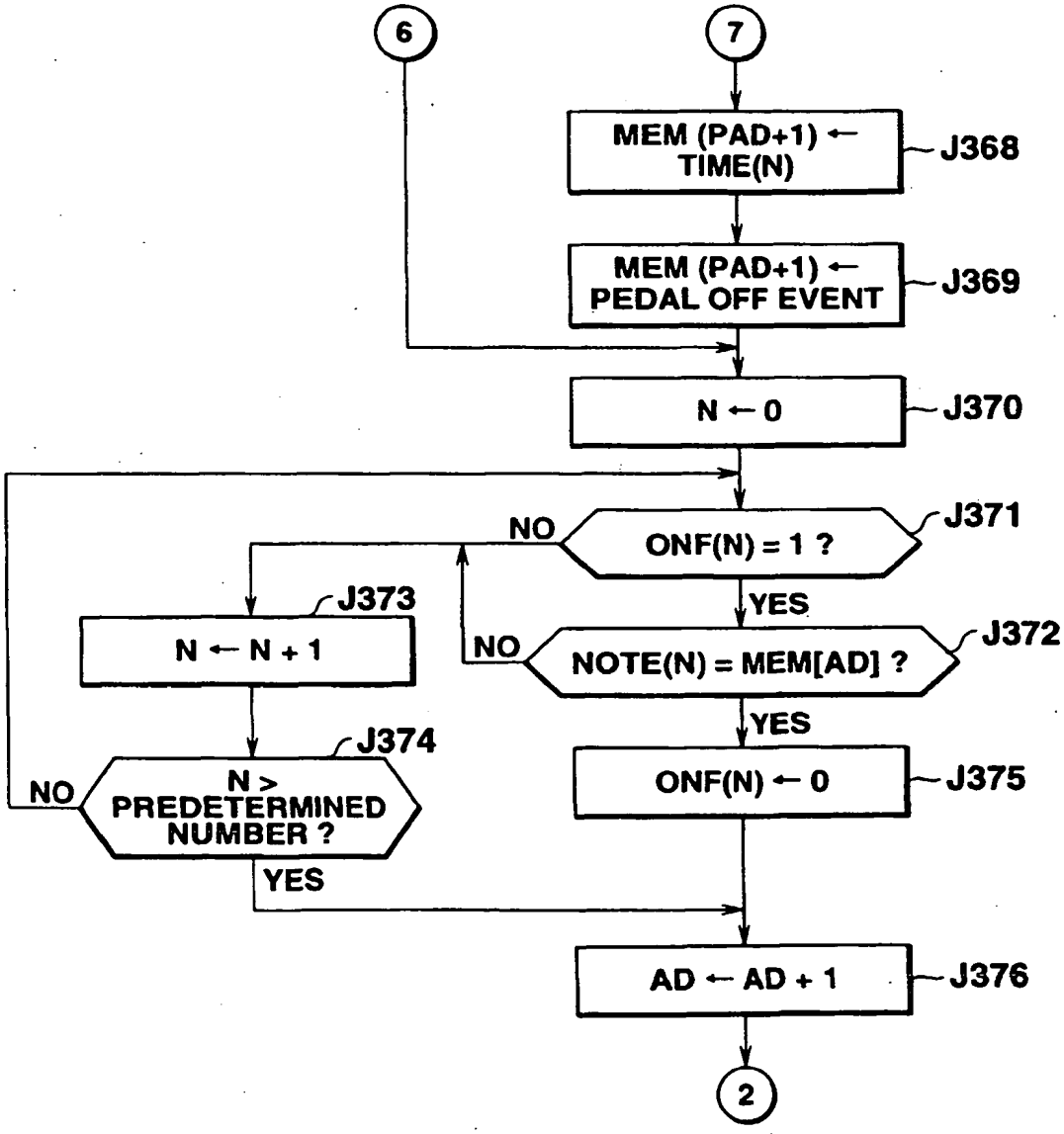
# FIG.42



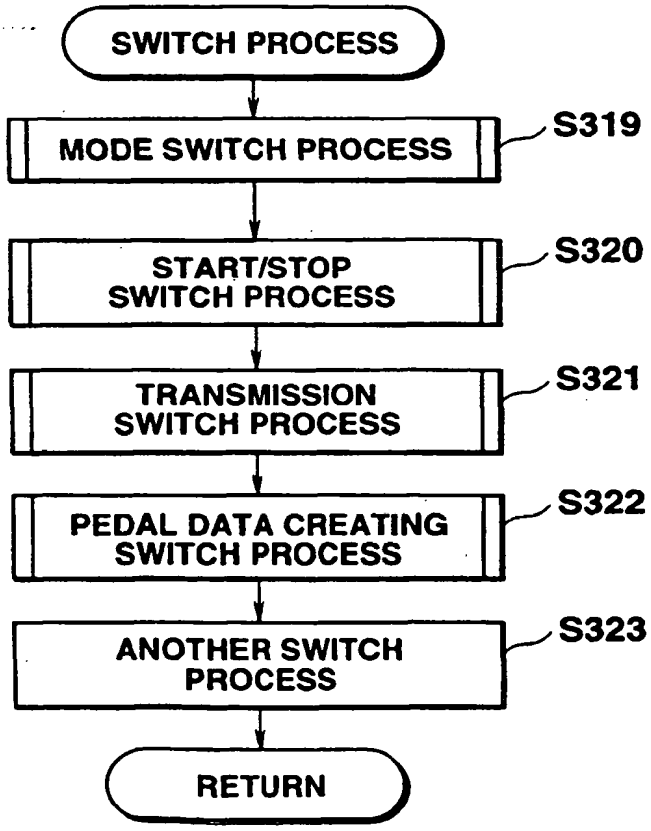
# FIG.44



# FIG.46



# FIG.48



# ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 99 10 2147

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

21-04-1999

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 3817144 A | 18-06-1974 | JP 952463 C | 25-05-1979 |
| | | JP 48094538 A | 05-12-1973 |
| | | JP 52024898 B | 04-07-1977 |
| US 5252775 A | 12-10-1993 | JP 3239292 A | 24-10-1991 |
| | | JP 8016840 B | 21-02-1996 |
| US 5589947 A | 31-12-1996 | JP 6102890 A | 15-04-1994 |

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82